

1. First we need to create users like sellers / normal users where sellers can post products i.e. create / add a new product to the application.

http://localhost:8080/ No Environment

POST http://localhost:8080/users Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "firstName": "Ashu",
3   "lastName": "Harshwardhan",
4   "email": "ashu.sae@gmail.com",
5   "role": 0,
6   "password": "password@123"
7 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 103 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 1,
3   "firstName": "Ashu",
4   "lastName": "Harshwardhan",
5   "email": "ashu.sae@gmail.com",
6   "role": "user",
7   "password": "password@123"
8 }
```

http://localhost:8080/ No Environment

POST http://localhost:8080/users Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "firstName": "Amit",
3   "lastName": "Yadav",
4   "email": "amit.yadav@gmail.com",
5   "role": 1,
6   "password": "password@456"
7 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 42 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 2,
3   "firstName": "Amit",
4   "lastName": "Yadav",
5   "email": "amit.yadav@gmail.com",
6   "role": "seller",
7   "password": "password@456"
8 }
```

- Next, we will create a category and under the category we will add respective products using the seller account.

POST http://localhost:8080/categories

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "categoryName": "Electronics",
3   "description": "All Electronics may come under this section",
4   "imageUrl": "/some/fake/url"
5 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 140 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 1,
3   "categoryName": "Electronics",
4   "description": "All Electronics may come under this section",
5   "imageUrl": "/some/fake/url",
6   "products": null
7 }
```

POST http://localhost:8080/categories

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "categoryName": "Mobile",
3   "description": "All Mobile may come under this section",
4   "imageUrl": "/some/fake/url/Mobile"
5 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 42 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 2,
3   "categoryName": "Mobile",
4   "description": "All Mobile may come under this section",
5   "imageUrl": "/some/fake/url/Mobile",
6   "products": null
7 }
```

3. Next, we will add products to these categories.

The image displays two screenshots of a REST client interface, showing the process of adding products to a database.

First Screenshot:

- Method:** POST
- URL:** http://localhost:8080/products
- Body:** JSON (application/json)
- JSON Body:**

```
1 {
2   "name": "Sony 55 inch TV",
3   "imageUrl": "/some/fake/url/sony",
4   "price": 1000,
5   "description": "Description of Sony 55 inch TV",
6   "categoryId": 1
7 }
```
- Status:** 201 Created
- Time:** 144 ms

Second Screenshot:

- Method:** POST
- URL:** http://localhost:8080/products
- Body:** JSON (application/json)
- JSON Body:**

```
1 {
2   "name": "Sony 21 inch TV",
3   "imageUrl": "/some/fake/url/sony21",
4   "price": 300,
5   "description": "Description of Sony 21 inch TV",
6   "categoryId": 1
7 }
```
- Status:** 201 Created
- Time:** 41 ms

POST

http://localhost:8080/products

Params

Send

Save

AuthorizationHeaders (1)BodyPre-request ScriptTestsCode

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"name": "iphone 11",

3

"imageUrl": "/some/fake/url/iphone11",

4

"price": 800,

5

"description": "Description of iphone11",

6

"categoryId": 2

7

}

BodyCookiesHeaders (5)Test ResultsStatus: 201 CreatedTime: 43 ms

PrettyRawPreviewJSON

1

{

2

"id": 3,

3

"name": "iphone 11",

4

"imageUrl": "/some/fake/url/iphone11",

5

"price": 800,

6

"description": "Description of iphone11"

7

}

POST

http://localhost:8080/products

Params

Send

Save

AuthorizationHeaders (1)BodyPre-request ScriptTestsCode

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"name": "iphone 12",

3

"imageUrl": "/some/fake/url/iphone12",

4

"price": 1500,

5

"description": "Description of iphone12",

6

"categoryId": 2

7

}

BodyCookiesHeaders (5)Test ResultsStatus: 201 CreatedTime: 40 ms

PrettyRawPreviewJSON

1

{

2

"id": 4,

3

"name": "iphone 12",

4

"imageUrl": "/some/fake/url/iphone12",

5

"price": 1500,

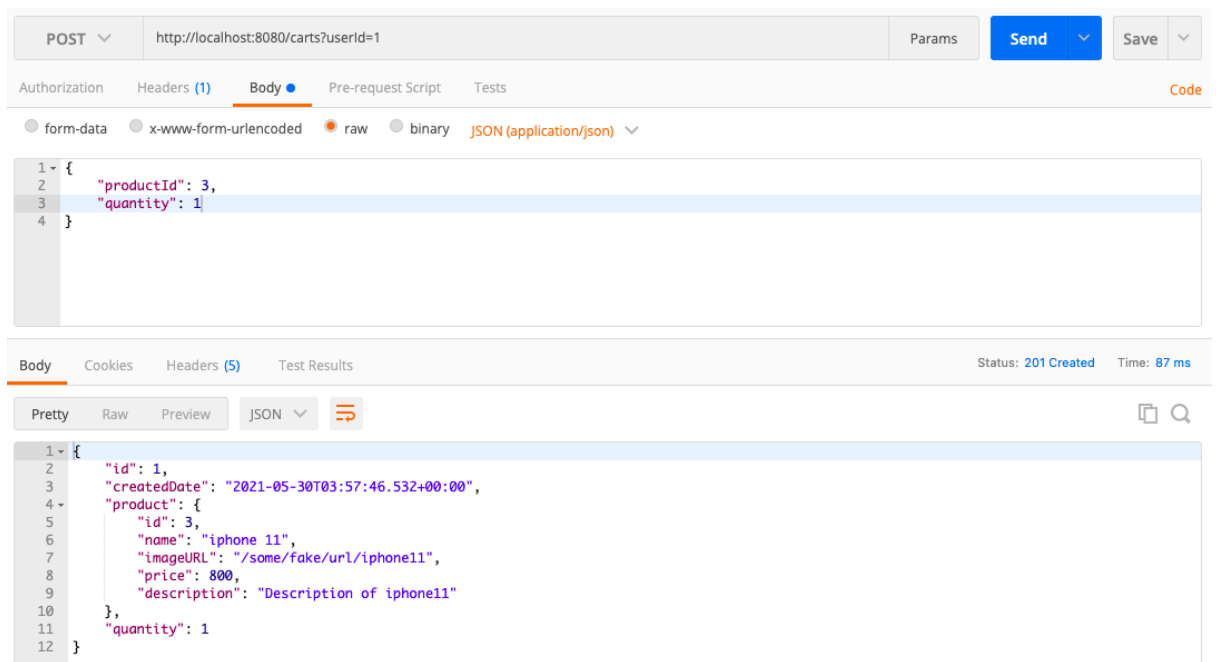
6

"description": "Description of iphone12"

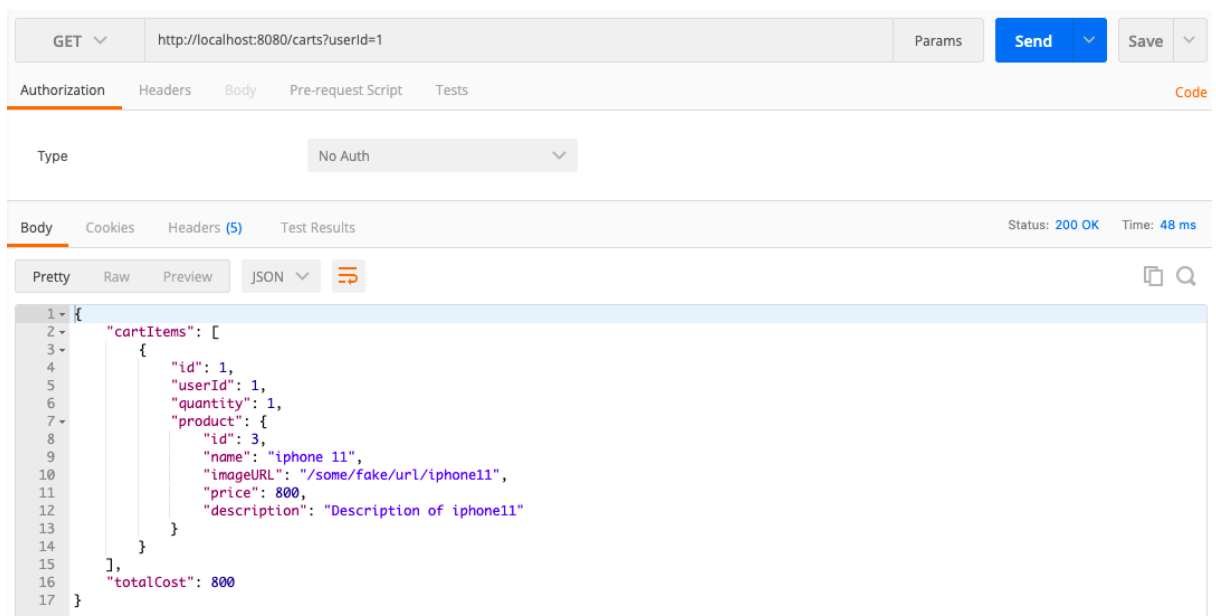
7

}

- Next, a normal user may add a product from a list of products and specify quantity and add that to his cart.



- At this point of time, if the purchaser wants to view the cart and see details, we can execute a GET call to get details.



6. Let's add another product to our cart.

POST http://localhost:8080/carts?userId=1

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "productId": 1,
3   "quantity": 1
4 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 53 ms

Pretty Raw Preview JSON

```
1 {
2   "id": 2,
3   "createdDate": "2021-05-30T04:04:32.768+00:00",
4   "product": {
5     "id": 1,
6     "name": "Sony 55 inch TV",
7     "imageUrl": "/some/fake/url/sony",
8     "price": 1000,
9     "description": "Description of Sony 55 inch TV"
10  },
11   "quantity": 1
12 }
```

7. Now check cart details using the GET call again.

GET http://localhost:8080/carts?userId=1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 49 ms

Pretty Raw Preview JSON

```
1 {
2   "cartItems": [
3     {
4       "id": 2,
5       "userId": 1,
6       "quantity": 1,
7       "product": {
8         "id": 1,
9         "name": "Sony 55 inch TV",
10        "imageUrl": "/some/fake/url/sony",
11        "price": 1000,
12        "description": "Description of Sony 55 inch TV"
13      }
14    },
15    {
16      "id": 1,
17      "userId": 1,
18      "quantity": 1,
19      "product": {
20        "id": 3,
21        "name": "iphone 11",
22        "imageUrl": "/some/fake/url/iphone11",
23        "price": 800,
24        "description": "Description of iphone11"
25      }
26    }
27  ],
28   "totalCost": 1800
29 }
```

8. Finally, the purchaser can make payment and after payment, the order will be placed and the application will call POST API to place the order.